TEXTBOOK MULTIGRID EFFICIENCY FOR THE STEADY EULER EQUATIONS

Thomas W. Roberts[*]
NASA Langley Research Center
David Sidilkover[†]
Institute for Computer Applications in Science and Engineering
and R. C. Swanson[*]
NASA Langley Research Center
Hampton, VA

Abstract

A fast multigrid solver for the steady incompressible Euler equations is presented. Unlike time-marching schemes, this approach uses relaxation of the steady equations. Application of this method results in a discretization that correctly distinguishes between the advection and elliptic parts of the operator, allowing efficient smoothers to be constructed. Solvers for both unstructured triangular grids and structured quadrilateral grids have been written. Computations for channel flow and flow over a nonlifting airfoil have computed. Using Gauss-Seidel relaxation ordered in the flow direction, textbook multigrid convergence rates of nearly one order-of-magnitude residual reduction per multigrid cycle are achieved, independent of the grid spacing. This approach also may be applied to the compressible Euler equations and the incompressible Navier-Stokes equations.

Introduction

One of the critical needs in computational fluid dynamics is faster flow solvers. Multigrid is a well known method of convergence acceleration which is widely used in Euler and Reynolds-averaged Navier-Stokes codes. These applications of multigrid generally are based on the unsteady equations using some temporal integrator as the smoother, combined with a full-approximation scheme (FAS) multigrid iteration. A common approach is one originally proposed by Jameson[1]. Starting with the unsteady equations, a finite-volume spatial discretization with explicit artificial viscosity is combined with a Runge-Kutta time integration as a smoother. An alternative approach[2, 3, 4] is to use upwind-differencing and implicit time integration as the smoother. However,

* Research Scientist
† Senior Staff Scientist

these approaches have resulted in poor multigrid efficiency. When applied to high Reynolds number flows over complex geometries, convergence rates are often worse than 0.99. There is clearly a need to develop substantially more efficient multigrid solvers.

According to Brandt[5], one of the major obstacles to achieving better multigrid performance for advection dominated flows is that the coarse grid provides only a fraction of the needed correction for smooth error components. This obstacle can be removed by designing a solver that effectively distinguishes between the elliptic, parabolic, and hyperbolic (advection) factors of the system and treats each one appropriately. For instance, advection can be treated by space marching, while elliptic factors can be treated by multigrid. The efficiency of such an algorithm will be essentially identical to that of the solver for the elliptic factor only, and thereby attain so-called "textbook" multigrid efficiency. Brandt presents an approach called "distributive relaxation" by which one can construct smoothers that effectively distinguish between the different factors of the operator. Using this approach, Brandt and Yavneh have demonstrated textbook multigrid for the incompressible Navier-Stokes equations[6]. Their results are for a simple geometry and a Cartesian grid, using a staggered-grid discretization of the equations.

In a closely related approach, Ta'asan[7] presents a fast multigrid solver for the compressible Euler equations. This method is based on a set of "canonical variables" which express the steady Euler equations in terms of an elliptic and a hyperbolic partition[8]. Ta'asan uses this partition to guide the discretization of the equations. A staggered grid is used, with different variables residing at cell, vertex, and edge centers. In Reference 7 it is shown that ideal multigrid efficiency can be achieved for the compressible Euler equations for two-dimensional subsonic flow using body-fitted grids. One possible limitation of the use of canonical variables is that the partition of the inviscid equations is not directly applicable to the viscous equations.

In this paper, an alternative to distributive relaxation and to Ta'asan's canonical variable decomposition is presented. It is a generalization of the approach

of Sidilkover and Ascher[9], and is described in detail by Sidilkover[10]. This approach can be classified as a method of the Weighted Gauss-Seidel type[5]. A conventional vertex-based finite-volume or finite-difference discretization of the primitive variables is used, avoiding the need for staggered grids. This simplifies the restriction and prolongation operations, because the same operator can be used for all variables. A projection operator is applied to the system of equations, resulting in a Poisson equation for the pressure. By applying the projection operator to the discrete equations rather than to the differential equations, the proper boundary condition on the pressure is satisfied directly. The Poisson equation for the pressure may be treated by Gauss-Seidel relaxation, while the advection terms of the momentum equation are treated by space-marching. Because the elliptic and advection parts of the system are decoupled, ideal multigrid efficiency can be achieved. Compared to distributive relaxation and the canonical variables approaches, this method is extremely simple.

Formulation of the Problem

The incompressible Euler equations in primitive variables are

$$u u_x + v u_y + p_x = 0,$$
$$u v_x + v v_y + p_y = 0,$$
$$u_x + v_y = 0,$$

where u and v are the components of the velocity in the x and y directions, respectively, and p is the pressure. The density is taken to be one. The advection operator is defined by

$$Q \equiv u \partial_x + v \partial_y, \tag{1}$$

where ∂_x, ∂_y are the partial differentiation operators. The Euler equations may be written as

$$\mathbf{L q} = \begin{pmatrix} Q & 0 & \partial_x \\ 0 & Q & \partial_y \\ \partial_x & \partial_y & 0 \end{pmatrix} \begin{pmatrix} u \\ v \\ p \end{pmatrix} = 0. \tag{2}$$

Introducing the adjoint to Q, defined by

$$Q^*(f) \equiv -\partial_x(uf) - \partial_y(vf), \tag{3}$$

a projection operator $\mathbf{P}$ is defined:

$$\mathbf{P} = \begin{pmatrix} I & 0 & 0 \\ 0 & I & 0 \\ \partial_x & \partial_y & Q^* \end{pmatrix}. \tag{4}$$

Applying the projection operator to the Euler equations yields

$$\tilde{\mathbf{L}} \mathbf{q} \equiv \mathbf{P L q} = \begin{pmatrix} Q & 0 & \partial_x \\ 0 & Q & \partial_y \\ 0 & 0 & \Delta \end{pmatrix} \begin{pmatrix} u \\ v \\ p \end{pmatrix} + \text{s.p.t.}, \tag{5}$$

where Δ is the Laplacian. The matrix operator on the right-hand side consists of the principal part of $\tilde{\mathbf{L}}$,

and "s.p.t." are the subprincipal terms, in the terminology of Brandt[5]. These terms arise because the coefficients u and v in the operators Q and Q^* are not constant. It is important to note that the subprincipal terms can be ignored for the purpose of constructing a relaxation scheme.

A feature of Eq. (5) is that the correct boundary condition for the pressure at a solid wall is an automatic result of the tangency boundary condition on the velocity. The equation for the pressure at the wall reduces to

$$(-u \partial_y v + v \partial_y u + \partial_x p) \hat{n}_x$$
$$+ (u \partial_x v - v \partial_x u + \partial_y p) \hat{n}_y = 0, \tag{6}$$

where $\hat{n}_x$, $\hat{n}_y$ are the components of the unit normal at the wall. Because the normal component of velocity, $u \hat{n}_x + v \hat{n}_y$, is zero, it can be shown that Eq. (6) reduces to the momentum equation in the normal direction. No auxiliary boundary condition on the pressure is needed.

The operator on the left-hand side of Eq. (5) is upper triangular. The pressure satisfies a Poisson equation for which a conventional relaxation method, such as Gauss-Seidel, can be applied. Upwind differencing of the advection operator in the momentum equations allows downstream relaxation to be used. The strategy used to relax the system is to first update the pressure. The pressure update contributes to the velocity update through the gradient terms in the right-hand column of the operator in Eq. (5). Finally, the velocity components are updated by relaxing in the stream direction.

Solution Procedure

Rather than discretizing Eq. (5) directly, the first step of numerical solution procedure is to discretize Eq. (2). The relaxation scheme is constructed by applying the projection operator $\mathbf{P}$ at the discrete level rather than the differential level. A sequence of grids $G_K, G_{K-1}, \dots, G_0$ is used, where G_K is the finest and G_0 the coarsest. Let $\tilde{\mathbf{L}}_k$ be the discrete approximation to the operator $\tilde{\mathbf{L}}$ and $\mathbf{q}_k$ be the solution on the k-th grid. This system has the form $\tilde{\mathbf{L}}_k \mathbf{q}_k = \mathbf{f}_k$, where the entries of $\tilde{\mathbf{L}}_k$ are 3×3 block matrices which operate on the unknowns $(u, v, p)^T$ at each grid vertex. A general iteration scheme is constructed by writing the operator $\tilde{\mathbf{L}}_k$ as $\tilde{\mathbf{L}}_k = \mathbf{M}_k - \mathbf{N}_k$, where the splitting is chosen such that $\mathbf{M}_k$ is easily inverted. Lexicographic Gauss-Seidel is obtained by taking $\mathbf{M}_k$ to be the block lower-triangular matrix resulting from ignoring the blocks above the diagonal of $\tilde{\mathbf{L}}_k$. A further simplification is obtained if the diagonal blocks of $\mathbf{M}_k$ contain only those entries corresponding to the principal part of the operator. Because the operator in Eq. (5) is upper triangular, the diagonal blocks will then be 3×3 upper triangular matrices.

Letting $\mathbf{q}_k^n$ be the n-th iterate of the solution, the iteration is

$$\mathbf{M}_k \mathbf{q}_k^{n+1} = \mathbf{f}_k + \mathbf{N}_k \mathbf{q}_k^n.$$

Because the operator $\tilde{\mathbf{L}}_k$ is nonlinear, $\mathbf{M}_k$ and $\mathbf{N}_k$ will be functions of $\mathbf{q}_k^n$ and $\mathbf{q}_k^{n+1}$. Letting $\delta\mathbf{q}_k^n \equiv \mathbf{q}_k^{n+1} - \mathbf{q}_k^n$, the iteration may be rewritten as

$$\mathbf{M}_k \delta\mathbf{q}_k^n = \mathbf{f}_k - \tilde{\mathbf{L}}_k \mathbf{q}_k^n. \tag{7}$$

Because $\mathbf{M}_k$ is block lower-triangular, $\delta\mathbf{q}_k^n$ is found by forward substitution and inverting a 3×3 diagonal block at each vertex. The diagonal blocks are upper triangular and are easily inverted.

If upwind differences are used for the advection operator (1) and the grid points are ordered in the flow direction, then the 3×3 blocks of $\mathbf{N}_k$ will have zeroes in the first two rows. In this case, lexicographic Gauss-Seidel relaxation is equivalent to space-marching of the advection terms. The advected error is effectively eliminated in one relaxation sweep and the convergence rate becomes that of the Poisson equation for the pressure. It is possible to get ideal multigrid convergence rates for the system because each component of the error is treated appropriately.

A straightforward FAS multigrid iteration is applied to the system of equations. Let $\tilde{\mathbf{L}}_{k-1}$ be the coarse grid operator, I_{k-1}^k be the fine-to-coarse grid restriction operator, and I_k^{k-1} be the coarse-to-fine grid prolongation operator. If $\hat{\mathbf{q}}_k$ is the current solution on grid k, the residual on this grid is $\mathbf{r}_k \equiv \mathbf{f}_k - \tilde{\mathbf{L}}_k \hat{\mathbf{q}}_k$. This leads to the coarse-grid equation

$$\tilde{\mathbf{L}}_{k-1}\mathbf{q}_{k-1} = \mathbf{f}_{k-1} = I_{k-1}^k \mathbf{r}_k + \tilde{\mathbf{L}}_{k-1}\left(I_{k-1}^k \hat{\mathbf{q}}_k\right). \tag{8}$$

After solving the coarse-grid equation for $\mathbf{q}_{k-1}$, the fine-grid solution is corrected by

$$\mathbf{q}_k \leftarrow \hat{\mathbf{q}}_k + I_k^{k-1}\left(\mathbf{q}_{k-1} - I_{k-1}^k \hat{\mathbf{q}}_k\right). \tag{9}$$

Equation (8) is solved by applying the same relaxation procedure that is used to solve the fine-grid equation. Multigrid is applied recursively to the coarse-grid equation. On the coarsest grid, many relaxation sweeps are performed to insure that the equation is solved completely. A conventional V–cycle or W–cycle is used.

Results

Two flow solvers have been written to implement the solution algorithm presented in the previous section. One is an unstructured grid code based on triangular elements. The program is a modification of the finite-volume code FUN2D[11, 12]. The discretization of the momentum equations uses the advection scheme proposed by Giles, et al.[13] The residuals of the momentum equations are computed by a trapezoidal rule integration around each triangle of the grid. These residuals are distributed to the nodes of the triangle using upwind-biased weights that depend on the local flow direction. One can think of this as a rotated-difference stencil. The Poisson equation for the pressure is discretized using a standard central-difference approximation for the Laplacian.

The other code is a structured, quadrilateral grid solver. A cell-vertex, finite-volume method is used at
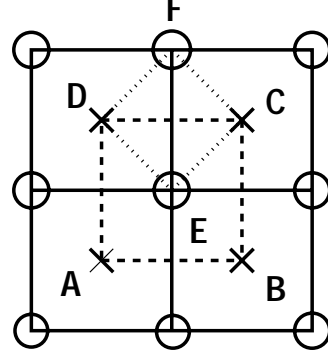

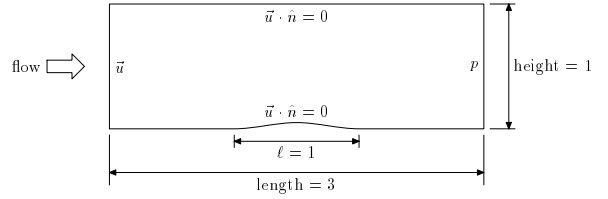Figure 1. Stencil for structured quadrilateral grid.

Figure 2. Channel geometry.

each grid point. The finite-volume cell associated with each vertex of the grid is the corresponding dual grid cell, as shown in Fig. 1. For vertex E the dual cell is $ABCD$. The advection terms are discretized by a conventional one-sided, second-order difference approximation. The gradients of u, v, and p appearing in the pressure Poisson equation are computed by applying Green's theorem. The gradient on the cell face CD of Fig. 1 is computed by integrating around the path $ECFD$. Derivatives of these gradients are also evaluated by Green's theorem. This approach yields the standard five-point approximation to the Laplacian operator on a uniform grid.

Solutions for incompressible, inviscid flow in a channel have been obtained with both solvers. The channel geometry and boundary conditions are shown in Fig. 2. The shape of the lower wall between $0 \le x \le 1$ is $y(x) = \tau \sin^2 \pi x$. For the computations shown here, the thickness ratio τ is 0.05. The velocity is specified at the inlet and the pressure is specified at the outlet. At the upper and lower walls of the channel, the flow tangency condition $\vec{u} \cdot \hat{n} = 0$ is enforced.

Similar grids were used for both flow solvers. Quasi-uniform quadrilateral grids were generated. A simple shearing transformation was used in the center part of the channel to get boundary conforming grids. For the unstructured grid solver, the grids were triangulated by dividing each quadrilateral cell along a diagonal. A series of nested coarse grids was obtained by coarsening the fine grids by a factor of two in each coordinate direction. In all cases shown below, the coarsest grid was 7×3 vertices. Lexicographic Gauss-Seidel relaxation was used. The ordering of the grid vertices was from the lower-left to the upper-right of the channel, resulting in downstream re-

laxation of the advection operator. A $V(2,1)$ multigrid cycle was used; that is, two relaxation sweeps were performed on each grid before restricting to the coarse grid, and one relaxation sweep was performed after the coarse-grid correction was added to the fine-grid solution. If one work unit is taken to be a single Gauss-Seidel sweep on the finest grid, a single $V(2,1)$ cycle is approximately 4 work units.

The computed pressure obtained by the unstructured grid flow solver on a grid of 97×33 vertices is shown in Fig. 3. A total of 5 grid levels was used. The L_1 norm of the residual of each equation for each cycle is shown in Fig. 4. The convergence rate is approximately 0.18 per
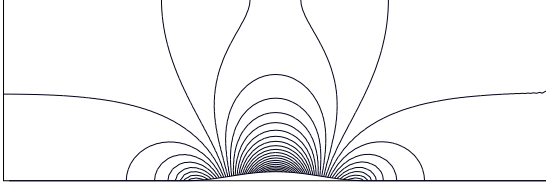
Figure 3. Pressure, contour increment $\Delta p = 0.01$, for an unstructured grid of 97×33 vertices.

multigrid cycle. For Gauss-Seidel relaxation applied to the Laplace equation, one can expect an optimum rate of 0.125 for a $V(2,1)$ multigrid cycle if the restriction and prolongation operators are perfect, so the current scheme gives very nearly the optimum rate. Also the convergence shows no sign of stalling but continues at a uniform rate. Changing to a W–cycle resulted in some improvement to the convergence rate per cycle, but required more work overall. (In two dimensions, a W–cycle is about 50% more work than a V–cycle.)

Very similar results were obtained by the structured grid flow solver as shown in Figs. 5 and 6. The convergence rate of the structured grid code is 0.15 per multigrid cycle, slightly better than that of the unstructured grid code and very close to the optimal rate. The better performance of the structured grid solver is most likely because of better restriction and prolongation operators;
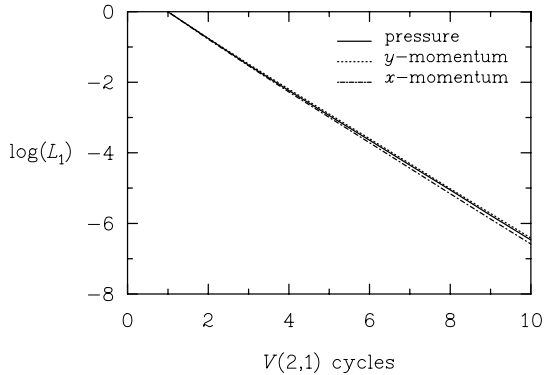

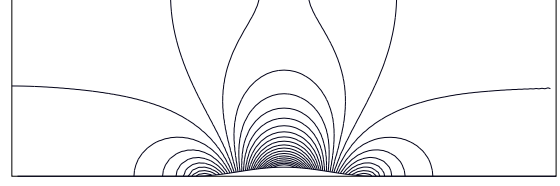Figure 4. Convergence history, unstructured grid of 97×33 vertices.

Figure 5. Pressure, contour increment $\Delta p = 0.01$, for a structured grid of 97×33 vertices.
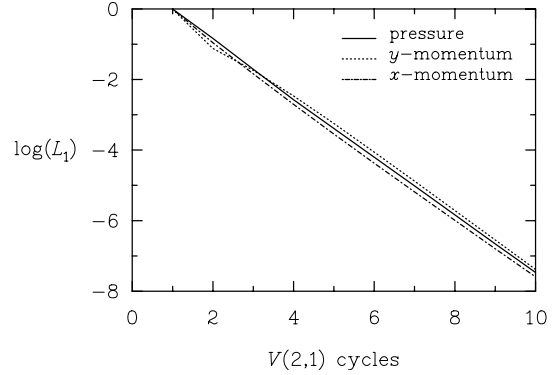

Figure 6. Convergence history, structured grid of 97×33 vertices.

the unstructured flow solver performs bilinear interpolation using only the locations of a fine-grid vertex and the three vertices of the coarse-grid cell containing that vertex.

The present scheme also shows ideal multigrid convergence independent of the grid spacing. In Fig. 7, the L_1 norm of the pressure residual is shown for the unstructured grid flow solver on a 49×17 fine grid using 4 grid levels, up to a 385×129 fine grid using 7 grid levels. A similar comparison for the structured grid flow solver is shown in Fig. 8. This shows that convergence is achieved in order n operations.
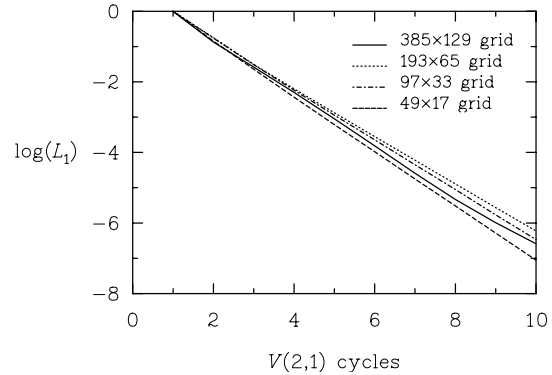

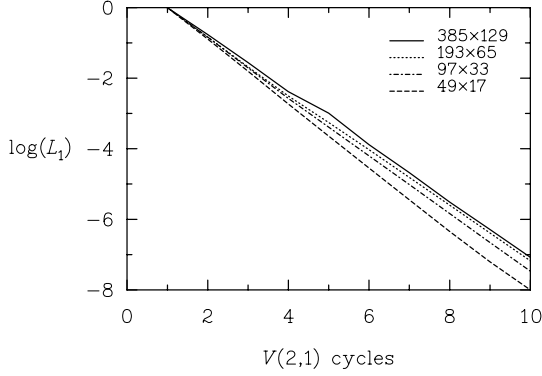Figure 7. Comparison of convergence rates on unstructured grids.

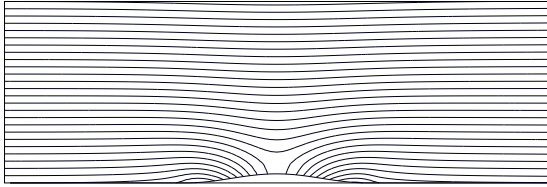Figure 8. Comparison of convergence rates on structured grids.

Figure 9. Shear flow, u-component of velocity, contour increment $\Delta u = 0.01$, unstructured grid of 97×33 vertices.

The flows shown so far are irrotational. To demonstrate that the multigrid performance does not degrade if vorticity is present, a solution for a shear flow in a channel was obtained. The unstructured grid flow solver was used for this computation. A linear shear was specified at the inlet; the velocity was $u(y) = (1+y)/2$, $v = 0$. The initial condition was a uniform flow in the channel. This insured that the vorticity would be introduced into the channel only by the relaxation. The u-velocity component is shown in Fig. 9 and the pressure is shown in Fig. 10. The convergence rate shown in Fig. 11 is seen to be essentially the same as in Fig. 4.

All the calculations shown so far were based on a series of nested grids: each coarse-grid vertex corresponds to a vertex on the next finer grid. For complex geometries it may not practical to generate a series of nested unstructured grids, and the performance of the multigrid solver may be expected to deteriorate. To show the robust-
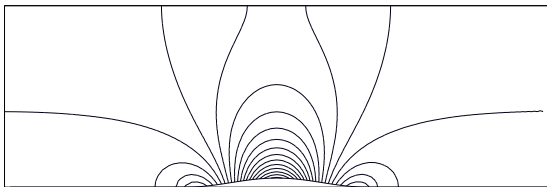

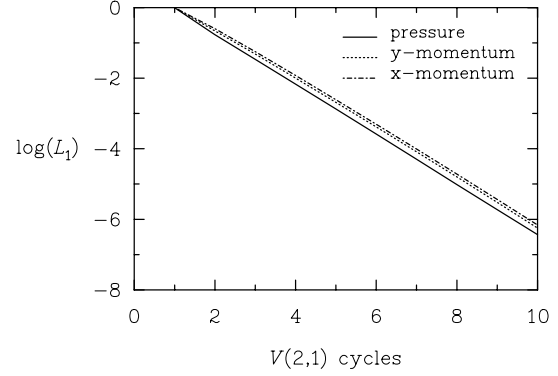Figure 10. Shear flow, pressure, contour increment $\Delta p = 0.01$, unstructured grid of 97×33 vertices.

Figure 11. Convergence rate, shear flow in a channel, unstructured grid of 97×33 vertices.
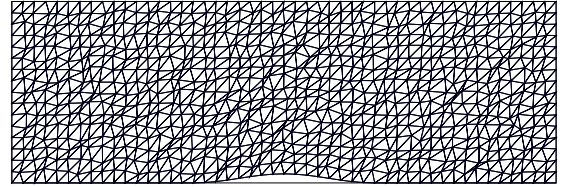

Figure 12. Grid generated by perturbing the vertices of the 49×17 grid.

ness of the current method, the triangular grid solver was run for a series of non-nested coarse grids. These were generated by randomly perturbing the locations of the vertices on each of the nested grids independently. The perturbed 49×17 grid is shown in Fig. 12. The computed pressure on a perturbed 97×33 fine grid with 5 grid levels is shown in Fig. 13 and the convergence rate is shown in Fig. 14. The pressure contours are very smooth, showing no sign of the lack of grid smoothness. The asymptotic convergence rate has deteriorated to a still-respectable 0.24 per cycle.

Solutions for incompressible, inviscid, nonlifting flow over a Kármán-Trefftz airfoil have been obtained with the structured grid solver. A cylinder is mapped into a Kármán-Trefftz airfoil using the conformal mapping

$$\frac{z-2}{z+2} = \left(\frac{\zeta - 1}{\zeta + 1}\right)^{2-\epsilon},$$
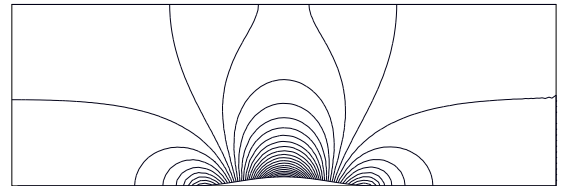

Figure 13. Pressure, contour increment $\Delta p = 0.01$, randomly perturbed unstructured grid of 97×33 vertices.
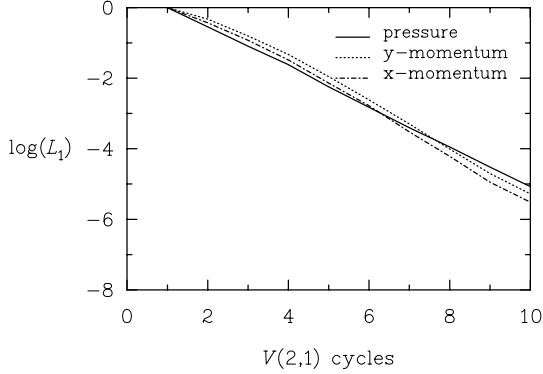
5

Figure 14. Convergence rate, randomly perturbed unstructured grid of 97×33 vertices, 5 grid levels, $V(2,1)$ cycle.
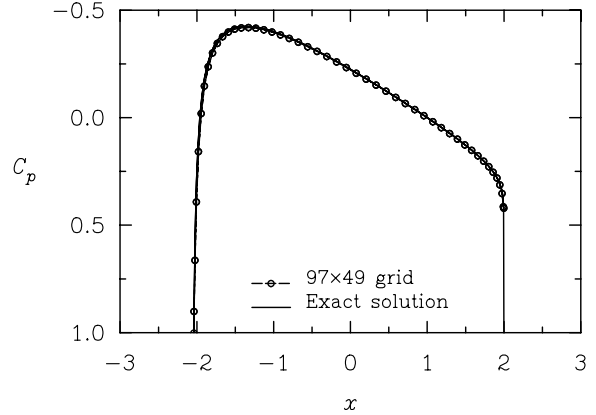

Figure 15. Pressure coefficient, nonlifting Kármán-Trefftz airfoil, 97×49 grid.

Figure 16. Surface pressure coefficient, nonlifting Kármán-Trefftz airfoil, 193×97 grid.

where z is the coordinate in the transformed plane, ζ is the coordinate in the circle plane, and ϵ is the trailing-edge angle divided by π. A trailing-edge angle of $10°$ is used, resulting in an airfoil of approximately 15% thickness. The airfoil flow is solved on a finite domain. At inflow points along the outer boundary the total pressure and flow inclination angle are specified. For outflow points the pressure is specified. The specified quantities are determined from the complex potential function for flow past the airfoil. On the airfoil surface the tangency condition is enforced.

A fine grid for the airfoil calculation was constructed by generating an O-grid with unit aspect ratio cells in the circle plane and mapping it to the airfoil plane. The outer boundary of the domain was roughly 13 chords from the airfoil. The coarsest grid in the grid sequence used for the multigrid solver contains 12×6 cells. On each grid, relaxation was performed along radial grid lines, sweeping from the inflow boundary to the airfoil surface over the forward half of the domain, and from the airfoil to the outflow boundary over the rearward half of the domain. Each sweep started along the stagnation streamline, proceeded over the upper half of the domain, and then over the lower half of the domain.

Comparisons between computed and analytic surface pressure coefficients for nonlifting flow around the Kármán-Trefftz airfoil are shown in Figs. 15 and 16. A $W(2,1)$ multigrid cycle was used for these computations. The computed surface pressure distribution for the 97×49 grid is in fairly good agreement with the analytic one. There are small differences near the leading edge and there is the usual difficulty in computing full pressure recovery at the trailing edge of the airfoil. Note that there is no clustering of the grid in these regions, which exacerbates the problem. The solution on the 193×97 grid is in excellent agreement with the analytic one, again except for the pressure recovery at the trailing edge. The computed pressure contours for the 193×97 are shown in Fig. 17.

The convergence history for the 193×97 grid is pre-

sented in Fig. 18. The L_1 norm of the residuals for all three equations is shown. The average rate of reduction of the residual is 0.12 per $W(2,1)$. A comparison of the convergence rates of the pressure residual for three grid densities is shown in Fig. 19. A slight deterioration of the convergence rate with increasing grid refinement is observed: on the 385×197 grid, the rate is 0.15 per cycles. Nevertheless, as with the channel flow results, the convergence rates are very nearly grid independent.
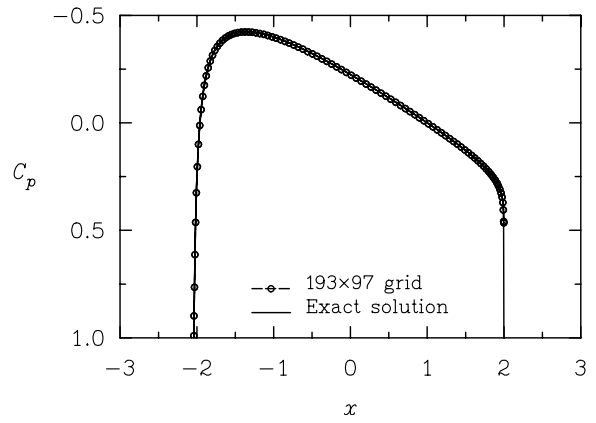
Extension to Compressible Flow

The general approach outlined here may be readily extended to the compressible Euler equation. In primitive variables the equations are

$$\mathbf{Lq} = \begin{pmatrix} Q & 0 & 0 & 0 \\ 0 & \rho Q & 0 & \partial_x \\ 0 & 0 & \rho Q & \partial_y \\ 0 & \rho \partial_x & \rho \partial_y & \frac{1}{c^2}Q \end{pmatrix} \begin{pmatrix} s \\ u \\ v \\ p \end{pmatrix} = 0. \qquad (10)$$
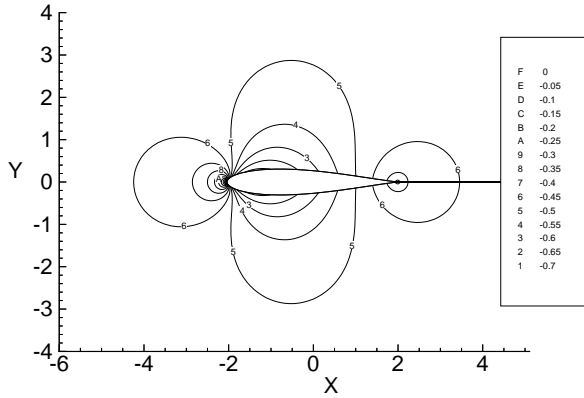
6

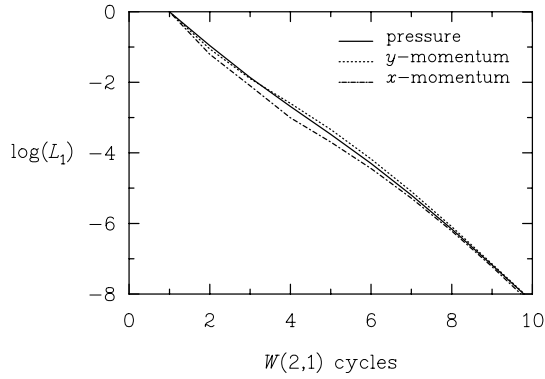Figure 17. Pressure coefficient contours, nonlifting Kármán-Trefftz airfoil, 193×97 grid.

Figure 18. Convergence history, nonlifting Kármán-Trefftz airfoil, 193×97 vertices.

where ρ is the density, c is the speed of sound, and s is the entropy. The projection operator $\mathbf{P}$ for this system is

$$\mathbf{P} = \begin{pmatrix} I & 0 & 0 & 0 \\ 0 & I & 0 & 0 \\ 0 & 0 & I & 0 \\ 0 & \partial_x & \partial_y & Q^* \end{pmatrix}. \qquad (11)$$

The operators Q and Q^* are defined as in Eqs. (1) and (3). Applying this to Eq. (10) and ignoring the subprincipal terms as before yields

$$\mathbf{PLq} = \begin{pmatrix} Q & 0 & 0 & 0 \\ 0 & \rho Q & 0 & \partial_x \\ 0 & 0 & \rho Q & \partial_y \\ 0 & 0 & 0 & \Delta - M^2 \partial_s^2 \end{pmatrix} \begin{pmatrix} s \\ u \\ v \\ p \end{pmatrix} + \text{s.p.t.}, \qquad (12)$$

where M is the Mach number and ∂_s is the partial derivative in the streamwise direction.

The most significant difference between the compressible and the incompressible equations is that a Prandtl-Glauert-like operator acts on the pressure. Note that
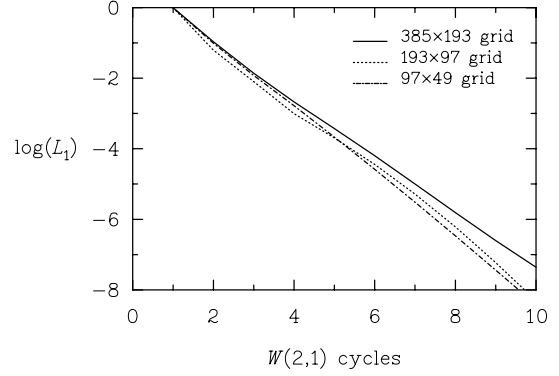

Figure 19. Comparison of convergence rates for non-lifting Kármán-Trefftz airfoil.

this system approaches the system for the incompressible equations in the limit of vanishing Mach number. For subsonic flow the compressible equations can be solved by the same relaxation scheme as the incompressible equations.

Although Eq. (10) is written in primitive variables, it is straightforward to use a conservative, finite-volume discretization of the system of equations. To relax the system, a transformation to primitive variables is made to compute the update of the solution values. For transonic flows with shocks a stable discretization can be achieved using the multidimensional high-resolution scheme developed by Sidilkover[14].

Conclusions

A multigrid algorithm which yields textbook multigrid efficiency for the steady Euler equations has been developed. It has the virtue of simplicity; conventional finite-difference or finite-volume discretizations of the governing equations may be used, allowing flexibility in the choice of the underlying numerical method. The correct boundary condition for the pressure equation is obtained directly. Unlike time-marching approaches, the speed of the method does not degrade for low-speed flows, and the correct incompressible limit is recovered. Finally, this method can be applied to incompressible, viscous flow following the ideas of Sidilkover and Ascher[9].

References

1. Jameson, A., "Solution of the Euler Equations for Two Dimensional Transonic Flow by a Multigrid

Method," *Appl. Math. Comput.,* vol. 13, nos. 3 and 4, pp. 327–355, 1983. University, 1983.

2. Mulder, W., "Multigrid Relaxation for the Euler Equations," *J. Comput. Phys.,* vol. 60, no. 2, pp. 235–252, 1985.

3. Anderson, W. K., Thomas, J. L., and Whitfield, D. L., "Three-Dimensional Multigrid Algorithms for the Flux-Split Euler Equations," NASA Technical Paper 2829, 1988.

4. Warren, G. P., and Roberts, T. W., "Multigrid Properties of Upwind-Biased Data Reconstructions," *Sixth Copper Mountain Conference on Multigrid Methods,* NASA Conference Publication 3224, Part 2, 1993.

5. Brandt, A., "Multigrid Techniques: 1984 Guide with Applications to Fluid Dynamics," GMD-Studie 85, GMD-FIT, 1985.

6. Brandt, A., and Yavneh, I., "Accelerated Multigrid Convergence and High-Reynolds Recirculating Flows," *SIAM J. Sci. Statist. Comput.,* vol. 14, no. 3, pp. 607–626, 1993.

7. Ta'asan, S., "Canonical-Variables Multigrid Method for Steady-State Euler Equations," ICASE Report 94–14, 1994.

8. Ta'asan, S., "Canonical Forms of Multidimensional Steady Inviscid Flow," ICASE Report 93–34, 1993.

9. Sidilkover, D., and Ascher, U. M., "A Multigrid Solver for the Steady State Navier-Stokes Equations using the Pressure-Poisson Formulation," *Comp. Appl. Math.* vol. 14, no. 1, pp. 21–35, 1995.

10. Sidilkover, D., "Some Approaches for Constructing Optimally Efficient Multigrid Solvers for the Inviscid Flow Equations," in preparation.

11. Anderson, W. K., and Bonhaus, D. L., "An Implicit Upwind Algorithm for Computing Turbulent Flows on Unstructured Grids," *Computers and Fluids,* vol. 23, no. 1, pp. 1–21, 1994.

12. Bonhaus, D. L., "An Upwind Multigrid Method for Solving Viscous Flows on Unstructured Triangular Meshes," M.S. thesis, George Washington University, 1993.

13. Giles, M., Anderson, W. K., and Roberts, T. W., "Upwind Control Volumes: A New Upwind Approach," AIAA Paper 90-0104, 1990.

14. Sidilkover, D., "A Genuinely Multidimensional Upwind Scheme and Efficient Multigrid Solver for the Compressible Euler Equations," ICASE Report 94–84, 1994.